Exemplar Based Non-parametric BRDFs

Antonio Haro, Irfan Essa
GVU Center / College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280

Abstract

Realistic rendering of computer modeled three dimensional surfaces typically involves building a parameterized model of the bidirectional reflectance distribution function (BRDF) of the desired surface material. We present a technique to render these surfaces with proper illumination and material properties using only a photograph of a sphere of the desired material under desired lighting conditions. Capitalizing on the fact that the geometry of the material in the photograph is known, we sample pixels of the sphere's reflectance to create photo-realistic renderings of computer models with the same material properties. The reflectance is sampled using texture synthesis techniques that compensate for the fact that very little of the BRDF observed in the photograph is known. The technique uses the limited observations of the function to create a plausible realistic rendering of the surface that can be composited onto a background plate easily.

1 Introduction

Rendering objects realistically has been one of the primary goals of computer graphics research for many years. Much progress has been made towards photo-realism as demonstrated by recent special effects in movies which are starting to become indistinguishable from real objects even to graphics experts. Typically, generating results that look photo-realistic requires knowledge of light-transport, accurate models of materials' bidirectional reflectance distribution functions (BRDFs), and good compositing software. This paper presents a technique for making untextured and unlit 3D models look perceptually realistic. The algorithm starts with a single photograph of a sphere made from the desired material. This photograph is then used to produce photo-realistic renderings of models from specified viewpoints as if they were made of the same material. We sample from the photograph of the material coherently, and with respect to the 3D surfaces of the source material and the model we want to render to produce a rendering from a desired viewpoint. This rendering is produced without any knowledge of surface material properties or scene lighting. Our technique in essence constructs a non-parametric approximation of the material's reflectance from the limited information in the single photograph our algorithm takes as input.

Our main contribution is in approximating the BRDF non-parametrically through the use of machine learning techniques. We never actually model the BRDF, but use the reflectance (with CCD sensor noise) present in our example material photograph to create our renderings of 3D models. As a result, we get very complex effects like multiple light source interactions and camera focus for free. In fact, for a single view of a 3D model, it is not necessary to know anything else about the material to produce realistic renderings with our technique. We start with a user posing a model in our 3D renderer to produce an image that captures the model's smooth 3D variations. Then, we photograph a sphere made of the desired material under the lighting conditions we want to emulate. A 3D wireframe sphere is aligned with the sphere in the photograph and is then rendered using our renderer. Finally, we use the three encoded normal "images" as input to the image analogies algorithm

Figure 1: The Stanford bunny rendered as chocolate lit from the right

[10] (with a very low setting for the coherence parameter) to produce a rendering of the model with the correct lighting and material properties needed to composite it into the source material's scene. Image analogies is used to approximate a very sparse function (reflectance produced by the material's BRDF is seen in only one image) by copying from the source photograph in a coherent manner.

We describe our approach in more detail in Section 3. The motivation and intuitions behind the algorithm are discussed in Section 3.1, and the specifics of our technique are in Section 3.2.

2 Previous Work

There is much previous work in rendering 3D models realistically. The subfields most related to our work are inverse global illumination, BRDF estimation, and texture synthesis. Our algorithm uses accomplishments in all of these areas to create realistic renderings.

Inverse global illumination involves using photographs to determine the surface properties and nature of lighting in the scene. Debevec [4] uses high dynamic range images of a chrome sphere to measure scene radiance and then uses the measured radiance to add new objects to the scene with correct lighting. These ideas were extended in the work by Yu *et al.*[19] where a number of high dynamic range photographs are used to determine the surface properties of objects in the scene as well. Our algorithm does not use a sphere to measure scene radiance. Rather, it uses a sphere to measure the radiance (plus the camera's CCD sensor noise) of the sphere's material we would like to use on 3D objects to add them to the scene with correct lighting.

Ramamoorthi and Hanrahan [13] also perform inverse global illumination by using a 3D model of an object and a photograph of the same object to recover the BRDF. The reflected light field is approximated with spherical harmonic basis functions which allow them to treat the problem of inverse rendering as a deconvolution to compute a parametric BRDF of the material in the photograph. Marschner *et al.*[12] use a set of photographs of an object to compute the object's BRDF. Each photograph's 3D camera position is computed, and then piped to a derenderer based on a raytracing renderer. For each pixel in each photograph, the radiance is divided by the irradiance to compute a BRDF value. All of the BRDF values then have a smooth, continuous function fit through them to yield a continuous BRDF over the entire domain. Our algorithm does not compute a parametric BRDF. Instead, it uses what is observed of the material's BRDF from a single image of a simple object to sample what little is known of the material's reflectance function smoothly.

Saito and Ikeuchi [14] take several photographs of an object whose BRDF they want to estimate under several lighting conditions and then approximate a parametric BRDF from the images. Debevec *et al.*[3] used a more dense sampling of images of faces under different illumination conditions to construct a reflectance function of human skin parameterized by the position of the camera and light position in each image. Since their sampling of the reflectance function is so dense, they do not have to perform any function fitting since they already have a parameterized approximation. Koudelka *et al.*[11] use a similar method to render surfaces with arbitrary BRDFs while handling illumination changes as well. Our data is not dense, rather, it is extremely sparse since we only have one photograph of the material under one unknown camera position and unknown lighting, so our method is based on treating each pixel of the material as an exemplar that is matched against coherently. Moreover, in these methods, the lighting must be known to produce realistic renderings. Our approach does not require any knowledge of scene lighting.

Texture synthesis is strongly related to our work and motivates our approach as well. The state of the art in texture synthesis is to model textures as Markov random fields (MRFs) [2, 5, 9] that are sampled to create new texture. MRF-based approaches build feature vectors composed of pixel neighborhoods in the input texture to build a feature space [16, 1]. This feature space is then matched against spatially to synthesize new texture. Recent work [1, 6] in texture synthesis implies that excellent texture synthesis results come from copying neighborhoods/patches from the input texture in a coherent manner. Our idea of copying neighborhoods from the photograph of the material we desire our 3D model to emulate is motivated from this observation. Recently, there has been work on doing texture synthesis in 3D over a model's surface [15, 18, 17]. While these methods produce seamless coverage of the entire surface, the BRDF of the input texture's material is not considered while matching over the surface. As a result, the textured model may not be lit realistically. Our method implicitly captures lighting effects while matching. We do not perform matching over the entire surface, but this option is considered in Section 6.

Our results appear similar to the 2D texture transfer results presented by Efros and Freeman [6]. However, the major difference is that we can achieve similar results on arbitrary 3D models instead of images. In addition, texture transfer over images of real objects is helped by the fact that the images of the real objects (the transfer source and destination) have already been lit. As a result, their algorithm implicitly uses specular highlights and lighting gradations over the photograph of the surface to help in matching. Their algorithm does not handle 3D models because the lighting in the scene would have to be known to light the model similarly to the object in the transfer source image. Our approach uses unlit, untextured 3D models and does not require the lighting in the scene to be known. Since our 3D models are unlit, we must simultaneously match the input material's albedo *and* lighting coherently over the surface of the model in the rendered view. Our approach is also similar to the texture by numbers application of image analogies presented in [10]. In the texture by numbers application, landscapes are the input so they can be approximated by gradient "planes". However, arbitrary 3D surfaces cannot be described so simply. Texture by numbers also requires a user to specify the "numbered" source image. In our algorithm, a user aligns an ellipsoid with the material sphere in our photograph, but this could be automated by using a greenscreen behind the material, or by doing background subtraction and removing the sphere's shadow, if any. The principles of our method are closest to those of Hamel and Strohotte [8]. In their work, a model is rendered with a non-photorealistic renderer, and then has its rendering "style" transferred to a new model by using various 2d representations of both 3D models such as curvature and shadows, as in G-buffers [14]).

3 Method

Our algorithm uses a digital photograph of a sphere made from the material that a complex 3D model shal resemble. Since spheres are such smoothly varying surfaces, we can fit 3D ellipsoids through the boundaries of the sphere photograph. Then, we encode the normals of our fitted 3D sphere and our 3D model as colors to produce two images; one of the fitted sphere, and one of the 3D model from the desired viewpoint. Finally, we sample from the photograph of the sphere material by using the two encoded normal images.

3.1 Motivation

The key intuition of our algorithm is that since spheres are smoothly varying, they can describe the majority of surfaces. That is, for 3D models that are smooth, virtually all surface variation can be explained by a single ellipsoidal hemisphere. We also assume that the light source is far away and that the 3D model does not shadow itself or surfaces in the scene (though recent work on producing realistic shadows for composited objects such as [7] addresses the latter).

A simple, but less effective, alternative to our approach is the following. Since we know the normals at every point on both surfaces (our source material and 3D model) we can use these to match reflectance values (from the photograph) to parts of the 3D model to create a photo-realistic rendering of the model. This simple alternative approach works well on source material that is untextured, such as clay. For source material that has texture, such as an orange, the simple approach will not yield good results. Problems arise if

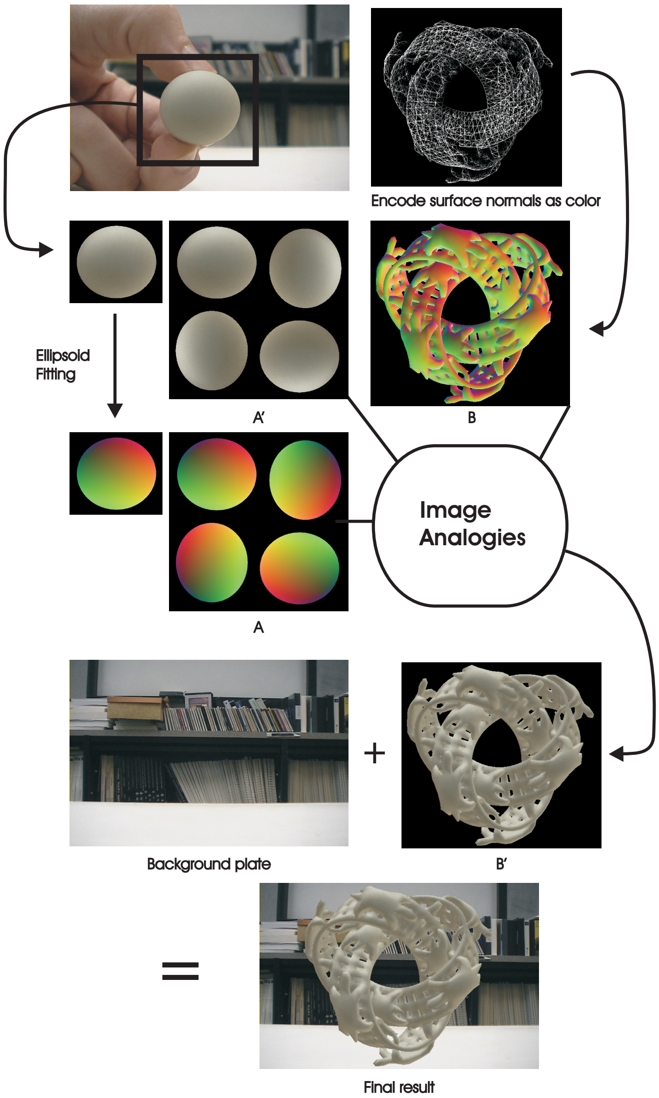


Figure 2: Our algorithm

the curvature of the surface is much lower or higher than that of the sphere. Portions with such curvature would look correctly lit with this approach, but would not have correct albedo since the sudden changes in curvature are not represented in the normal field of our source material.

Simple nearest neighbor distorts the albedo in curvaceous regions of the 3D model, so we enforce copying in regions where the curvature is significantly different than the source material. However, we do not want to over-copy since the sphere would then be visible in our rendered results. An algorithm which matches vectors by using nearest neighbors in the vector space and copies coherently from regions in the space when the nearest neighbor does not match previously matched vectors is image analogies [10].

Image analogies is used as our matching algorithm since it is simple nearest neighbors at low settings of its coherency parameter yet still copies from the source image coherently if the nearest neighbor would not be coherent with previously matched results. With coherent matching, our approach can handle textured materials without distorting the albedo when sampling.

3.2 Matching reflectance

The steps of our method are:

1. Render a 3D model with its normals encoded as colors along with a visibility mask

2. Render a 3D sphere with its normals also encoded as colors

3. Distort the rendered sphere image so that it lines up with the sphere in the photograph of the source material closely

4. Use image analogies to sample from the photograph of the material based on the two encoded normal renderings

5. Use the visibility mask to crop out any regions where the rendering does not match the silhouette of the 3D model

Figure 2 pictorially describes our pipeline. We render all 3D models with a simple renderer that produces images of the model at the current user specified pose with the normals encoded as color, i.e. (R,G,B) maps to (X,Y,Z) and the range (-1,1) maps to (0, 255). We perform this encoding so that we can perform matching over a 2D surface (the encoded normal image) instead of over the 3D surface itself (which we would then have to parameterize somehow).

A user aligns the rendered 3D sphere model with the photograph of the material sphere. This photograph is then rotated by 90 degrees three times to produce four pairs of rendered/real ellipsoids. These four pairs are used to make two images; one of the four real spheres, and one of the four rendered spheres. This is done because the 3D model's encoded normals may vary differently than the sphere's do.

We use the visibility mask produced by our renderer to crop out regions where image analogies does not follow the rendered object's silhouette. It is possible for the matching algorithm not to follow the silhouette perfectly because of the coherence matching. Once we apply our visibility mask, we can composite our rendered model onto a photograph of the source material sphere's scene without the sphere in it using any image manipulation program (e.g. Adobe Photoshop, Gimp, etc.)

4 Results

We have run our algorithm on many different combinations of lighting conditions, materials, and models. Figure 3 demonstrates that the algorithm is able to light the rendered models correctly without any knowledge of the lights in the scene. In addition, the results are very different for the two materials. This is because the method is implicitly approximating the reflectance equations for the materials while performing matching. As a result, the rendered models reflect light as their source materials do, thus making them appear realistic.

In Figure 4 we show a few results taken from objects that were photographed outside. The method produces images which composite very well into the background plate. In Figure 5 we show many results with different models, materials, and lighting conditions. The method produces realistic results for this large number of different and challenging combinations. Our approach gives the best results when the input material does not have a high-frequency albedo. This is because we do not have enough information to sample all of the high frequencies as well as diffuse and specular reflections. With additional viewpoints to sample from however, our method should achieve increasingly better results. Our method also performs well on complicated geometry with many holes. In addition, our method implicitly approximates the reflectance of materials with complicated BRDFs, such as beef, while matching to produce realistic renderings.

5 Conclusion

We have presented a method for rendering 3D models realistically that captures complex light/material interaction effects, without building a parametric model of the BRDF or using principles of light transport. Our method takes advantage of the fact that the geometry of a photographed sphere made from a material that we would like a 3D model to resemble is known. Since the geometry is known, we can then sample from the photograph in rendering a realistic image of our 3D model from a given viewing position.

We have presented results showing that our algorithm works for a large number of different material types, lighting conditions, and models. In addition, our algorithm implicitly captures light direction as well, without explicitly trying to figure out where light sources are or how many of them there are. Our method combines ideas from BRDF measurement, texture synthesis, and machine learning to create imagery that looks very realistic easily.

6 Future work

There are many exciting avenues of future research that we are interested in. We are interested in using this approach to produce animations. Currently, any animations produced by our approach would be temporally incoherent because temporal variations of the surface are not taken into account at all. We would also like to be able to move the models and light sources such that the models continue looking photo-realistic. This will require more information in the feature vector to encode other salient information such as light source positions. The matching algorithm will probably have to be modified as well so that specular highlights do not pop on or off the model's surface. It might also be useful to have additional information in the feature vector, such as distance to the camera (or other G-buffer properties), user specified constraints for highly textured material, or skeleton information of the model [8]. By making the feature vector more descriptive, its dimensionality will be increased, but with the introduction of multiple viewpoints such information could make it possible to use this algorithm as a basis for 3D interaction.

References

[1] Michael Ashikhmin. Synthesizing natural textures. *2001 ACM Symposium on Interactive 3D Graphics*, pages 217–226, March 2001.

[2] Jeremy S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. *Proceedings of SIGGRAPH 97*, pages 361–368, August 1997.

[3] P. Debevec, T. Hawkins, C. Tchou, H. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Computer Graphics (Proceedings of SIGGRAPH 2000)*, 145-156.

[4] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 189–198, Orlando, Florida, July 1998. ACM SIGGRAPH / Addison Wesley.

[5] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *ICCV99*, pages 1033–1038, 1999.

[6] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.

[7] Simon Gibson and Alan Murta. Interactive rendering with real world illumination. In *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 365–376. Eurographics, June 2000.

[8] J. Hamel and T. Strothotte. Capturing and re-using rendition styles for non-photorealistic rendering. *Computer Graphics Forum*, 18(3):173–182, September 1999.

[9] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. *Proceedings of SIGGRAPH 95*, pages 229–238, August 1995.

[10] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. *Proceedings of SIGGRAPH 2001*, pages 327–340, August 2001.

[11] M.L. Koudelka, P.N. Belhumeur, S. Magda, and D.J. Kriegman. Image-based modeling and rendering of surfaces with arbitrary brdfs. In *CVPR01*, pages I:568–575, 2001.

[12] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. Image-based brdf measurement including human skin. In *Eurographics Rendering Workshop 1999*, Granada, Spain, June 1999. Springer Wein / Eurographics.

[13] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 117–128. ACM Press / ACM SIGGRAPH, August 2001.

[14] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. In *Computer Graphics (Proceedings of SIGGRAPH 90)*, pages 197–206, Dallas, Texas, August 1990.

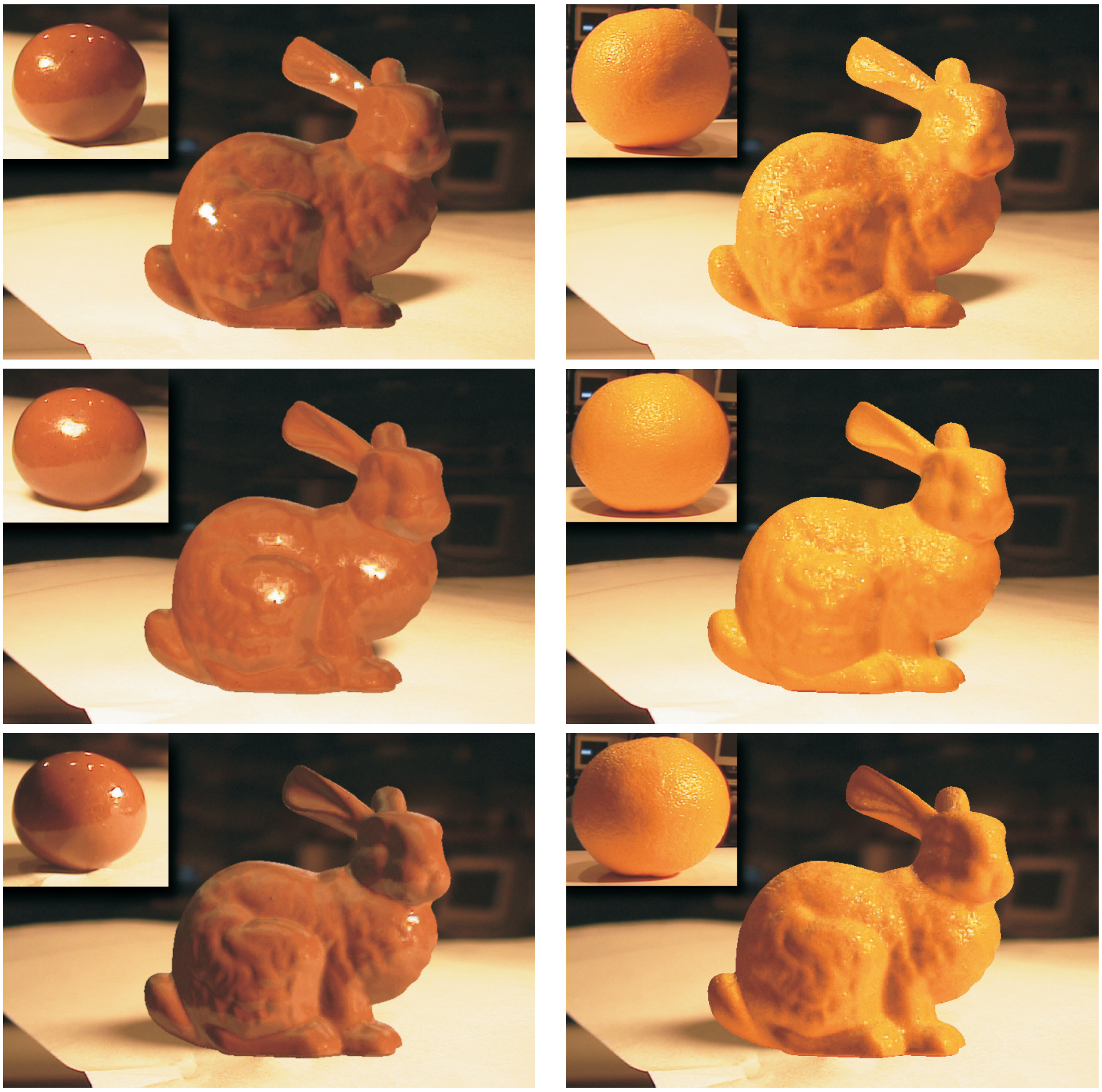
[15] Greg Turk. Texture synthesis on surfaces. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 347–354. ACM Press / ACM SIGGRAPH, August 2001.

[16] L. Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. *Proceedings of SIGGRAPH 2000*, pages 479–488, July 2000.

[17] Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 355–360. ACM Press / ACM SIGGRAPH, August 2001.

[18] Lexing Ying, Aaron Hertzmann, Henning Biermann, and Denis Zorin. Texture and shape synthesis on surfaces. In *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, pages 301–312. Eurographics, June 2001.

[19] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 215–224, Los Angeles, California, August 1999. ACM SIGGRAPH / Addison Wesley Longman.



(a) Chocolate **(b) Orange**

Figure 3: Renderings of the bunny using chocolate and orange as materials under varying illumination. The shadow is of the original object in the scene, not of the model. Inset in each image is the material photograph that was sampled to produce the rendering.

(a) Blue clay **(b) Mouse ball** **(c) White clay** **(d) Red clay**

Figure 4: Renderings of the bunny using various materials photographed outdoors.

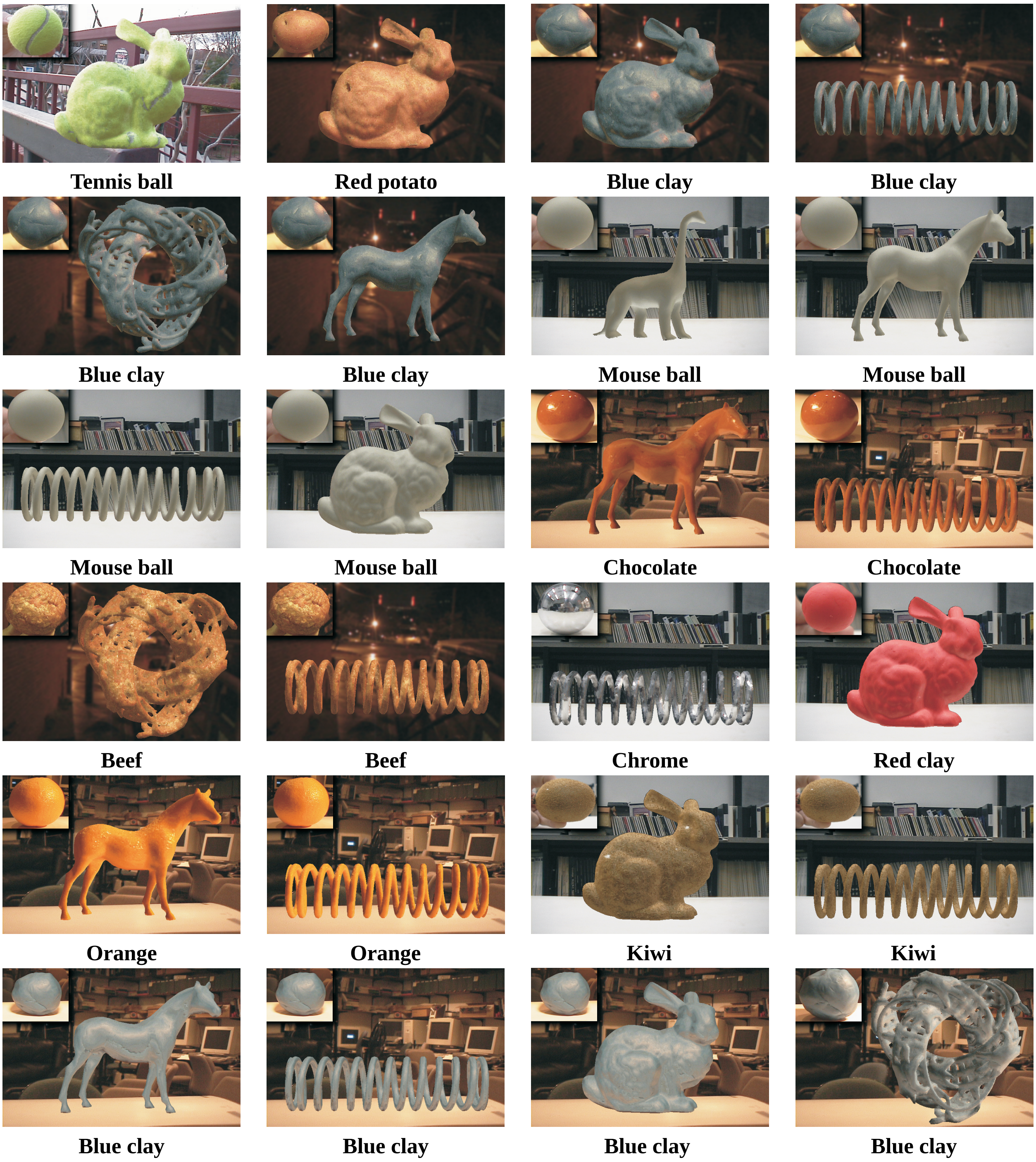


Figure 5: Results of our algorithm run on different materials, lighting conditions, and 3D models